

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Su-jeong LEE

Application No.: New

Group Art Unit: New

Filed: November 13, 2003

Examiner: New

For: STORAGE MEDIUM HAVING OBJECT-ORIENTED PROGRAM

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s) herewith a certified copy of the following foreign application:

Korean Patent Application No(s). 2003-10162

Filed: February 18, 2003

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing date(s) as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP



Date: November 13, 2003

By: _____

Michael D. Stein
Registration No. 37,240

1201 New York Ave, N.W., Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501
MDS/MQA/cmt



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출원 번호 : 10-2003-0010162
Application Number

출원 년 월 일 : 2003년 02월 18일
Date of Application FEB 18, 2003

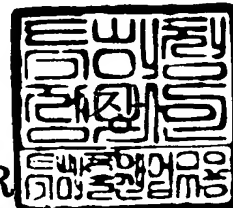
출원인 : 삼성전자주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2003 년 03 월 10 일

특 허 청

COMMISSIONER





【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【제출일자】	2003.02.18
【발명의 명칭】	객체 지향 프로그램이 기록된 저장 매체
【발명의 영문명칭】	Storage Medium Having Object-Oriented Program
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	허성원
【대리인코드】	9-1998-000615-2
【포괄위임등록번호】	2003-002172-2
【대리인】	
【성명】	윤창일
【대리인코드】	9-1998-000414-0
【포괄위임등록번호】	2003-002173-0
【발명자】	
【성명의 국문표기】	이수정
【성명의 영문표기】	LEE, SU JEONG
【주민등록번호】	690126-1122910
【우편번호】	442-725
【주소】	경기도 수원시 팔달구 영통동 972-2 벽적골 주공아파트 841동 904호
【국적】	KR
【심사청구】	청구
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규 에 의한 출원심사 를 청구합니다. 대리인 허성원 (인) 대리인 윤창일 (인)
【수수료】	
【기본출원료】	20 면 29,000 원
【가산출원료】	11 면 11,000 원



1020030010162

출력 일자: 2003/3/11

【우선권주장료】	0	건	0	원
【심사청구료】	13	항	525,000	원
【합계】	565,000			원

【요약서】**【요약】**

본 발명은, 임의의 데이터를 저장할 객체 변수 및 상기 객체 변수를 통하여 상기 데이터에 수행되는 복수의 함수를 포함하는 객체로 구성된 객체 지향 프로그램이 기록된 저장 매체에 관한 것이다. 본 객체 지향 프로그램이 기록된 저장 매체는, 객체 변수의 변수명을 선택적으로 저장할 수 있는 변수 식별자와, 상기 객체 변수에 저장될 데이터의 변수 타입을 선택적으로 저장할 수 있는 변수 타입 정보와, 상기 데이터가 저장되어 있는 메모리 주소를 지시하는 주소 정보를 포함할 수 있는 마스터 클래스와; 상기 마스터 클래스 내에 마련되어, 상기 변수 식별자, 변수 타입 및 데이터 값을 입력받아 상기 객체 변수에 상기 데이터가 저장될 수 있도록 하는 데이터 저장 함수를 포함하는 것을 특징으로 한다. 이에 의하여, 하나의 클래스로부터 다양한 형태의 객체를 동적으로 생성할 수 있어, 클래스 설계 및 프로그램 개발에 드는 시간 및 비용을 절감할 수 있다.

【대표도】

도 4



【명세서】

【발명의 명칭】

객체 지향 프로그램이 기록된 저장 매체{Storage Medium Having Object-Oriented Program}

【도면의 간단한 설명】

도 1은 본 발명의 바람직한 실시예에 따른 마스터 클래스의 구성도,

도 2는 본 발명의 바람직한 실시예에 따른 위치 정보를 포함하는 변수 식별자를 나타내는 도면,

도 3a 및 3b는 본 발명의 데이터 저장 함수를 적용하여 형성된 임의의 데이터 구조 예를 나타내는 도면,

도 4는 본 발명에 따른 데이터 저장 함수의 작동을 보여주는 흐름도,

도 5는 본 발명에 따른 데이터 조회 함수의 작동을 보여주는 흐름도,

도 6은 본 발명에 따른 문자열 변환 함수의 작동을 보여주는 흐름도이다.

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<7> 본 발명은 임의의 데이터를 저장할 객체 변수 및 상기 객체 변수를 통하여 상기 데이터에 수행되는 복수의 함수를 포함하는 객체로 구성된 객체 지향 프로그램이 기록된 저장 매체에 관한 것으로서, 보다 상세하게는 하나의 클래스로부터 다양한 형태의 객체를 동



적으로 생성할 수 있어, 클래스 설계 및 프로그램 개발에 드는 시간 및 비용을 절감할 수 있는 객체 지향 프로그램이 기록된 저장 매체를 제공하는 것이다.

<8> 기존의 절차 지향 프로그램(Procedural-Oriented Program)은 데이터 구조와 그 데이터를 변화시키는 알고리즘으로 구성된다. 이러한 절차 지향 프로그램은 데이터와 함수의 연관 관계를 잘 표현하지 못하므로, 알고리즘의 변경으로 인한 보수유지 및 소스코드의 재사용에 있어서 한계를 가지고 있다.

<9> 따라서, 보수유지 및 재사용이 용이한 프로그램 제작을 위해, 객체 지향 프로그램(Object-Oriented Program)이라는 개념이 창안되었는데, 이는 객체 고유의 데이터와 그 데이터를 변화시킬 수 있는 함수인 메소드로 구성된 객체들이 메세지(Message)를 통하여 상호 통신함으로써 원하는 결과를 얻을 수 있도록 구성된다.

<10> 여기서, 객체란 효율적인 정보관리를 위하여 의미를 부여하고 분류하는 개념적인 단위로서, 특정한 데이터와 관련된 함수들을 가지는 프로그램 모듈을 지칭하는 것이다. 데이터는 그 객체의 속성을 저장하는데 사용되고, 함수는 그 객체가 수행할 수 있는 기능을 정의한다. 각 객체는 속성을 저장하기 위해 최소한 한 개 이상의 변수를 가지고 있어야 한다. C++, Java, Python 등의 객체 지향 프로그래밍 언어에 있어서, 객체는 상기 객체의 공통된 특성 및 동작을 포함/수행할 수 있도록 틀을 제공해 주는 객체 클래스로부터 생성된다.

<11> 여기서, 각 객체들은 미리 정해진 변수 타입을 가지는 고정된 개수의 변수에 저장된 데이터를 가지는 것이 일반적이다. 그러므로, 변수 타입이 다르거나 변수 개수가 다른 경우에는 그 마다 각각의 클래스를 정의하여 사용해야 했다.



- <12> 그러므로, 다양한 종류의 객체를 하나의 클래스로부터 생성함으로써, 객체 지향 프로그래밍 언어의 원래 목적인 보수유지 및 재사용을 최대한 달성할 수 있도록 하는 기술에 대한 제안이 필요하다.
- <13> 한편, 객체 직렬화란 객체의 내용을 파일 또는 네트워크를 통하여 바이트 스트림으로 송수신하기 위하여 객체를 스트림화하는 것이다. 또한, 직렬화된 데이터에서 원래의 객체로 복원하는 것을 탈직렬화라고 한다. 이렇게 객체를 직렬화함으로써 얻을 수 있는 장점은 객체 자체의 내용을 입출력 형식에 구애받지 않고 객체를 파일에 저장함으로써 영속성(persistency)을 제공할 수 있고, 객체 자체를 네트워크를 통하여 손쉽게 교환할 수 있게 된다는 데 있다.
- <14> C++에서는 이러한 기능을 수행할 수 있도록 Object 클래스 내에 직렬화(serialize) 메소드를 마련하여 여러 가지 기본형과 객체들에 대한 직렬화를 수행할 수 있도록 지원해주고 있다. 따라서, 객체 직렬화를 위해서는 각 클래스에 직렬화 메소드를 오버라이드(override)해서 각 클래스의 특성에 맞게 커스터마이징(customizing)하는 작업이 수반되어야 하는 불편함이 있었다.

【발명이 이루고자 하는 기술적 과제】

- <15> 따라서, 본 발명의 목적은, 다양한 종류의 객체를 생성할 수 있도록 변수 타입이나 변수 개수에 한정되지 않는 새로운 하나의 클래스를 제공함으로써, 클래스 설계 및 프로그램 개발에 드는 시간 및 비용을 절감할 수 있는 객체 지향 프로그램이 기록된 저장매체를 제공하는 것이다.

【발명의 구성 및 작용】

- <16> 상기 목적은, 본 발명에 따라, 임의의 데이터를 저장할 객체 변수 및 상기 객체 변수를 통하여 상기 데이터에 수행되는 복수의 함수를 포함하는 객체로 구성된 객체 지향 프로그램이 기록된 저장 매체에 있어서, 객체 변수의 변수명을 선택적으로 저장할 수 있는 변수 식별자와, 상기 객체 변수에 저장될 데이터의 변수 타입을 선택적으로 저장할 수 있는 변수 타입 정보와, 상기 데이터가 저장되어 있는 메모리 주소를 지시하는 주소 정보를 포함할 수 있는 마스터 클래스와; 상기 마스터 클래스 내에 마련되어, 상기 변수 식별자, 변수 타입 및 데이터 값을 입력받아 상기 객체 변수에 상기 데이터가 저장될 수 있도록 하는 데이터 저장 함수를 포함하는 것에 의해 달성된다.
- <17> 여기서, 상기 변수명은, 문자열로 표현되는 것이 바람직하다.
- <18> 또한, 상기 변수 타입 정보는, 문자열로 표현되고, 해당 객체 변수가 적어도 하나의 데이터를 저장할 수 있는 다른 객체 변수를 지시할 수 있도록 마스터 클래스를 지정할 수 있는 것이 바람직하다.
- <19> 나아가, 상기 주소 정보는, 하나 이상의 데이터를 저장할 수 있도록 포인터의 리스트로 구성되는 것이 바람직하다.
- <20> 상기 데이터 저장 함수에서 입력받는 상기 변수 식별자는, 객체 내의 객체 변수들이 다른 객체 변수를 지시함으로써 형성되는 객체 간의 연결 관계인 소정의 데이터 구조상의 위치 정보를 포함하고 있고, 상기 데이터 저장 함수는 해당 위치의 객체 변수에 상기 데이터를 저장하는 것이 바람직하다.

- <21> 여기서, 상기 데이터 구조는 객체 변수간 연결에 따라 레이어가 증가되는 구조이고, 상기 위치 정보는 데이터를 저장하고자 하는 객체 변수에 계층적으로 연결된 상위 레이어에 위치한 다른 객체 변수의 변수 식별자와, 상기 객체 변수들의 변수 식별자를 각각 구분하는 구분자를 포함할 수 있다.
- <22> 또한, 상기 데이터 저장 함수는, 상기 데이터 구조상의 해당 위치에 객체 변수가 존재하는 경우, 변수 타입을 정수배열, 문자열배열과 같은 어레이 형식의 변수 타입으로 설정하고, 상기 객체 변수에 상기 데이터를 추가로 저장하는 것이 바람직하다.
- <23> 나아가, 상기 데이터 저장 함수는, 상기 위치 정보 중 각 레이어에 위치한 다른 객체 변수가 상기 데이터 구조상에 존재하지 않는 경우, 변수 식별자로 상기 존재하지 않는 객체 변수의 식별자를, 변수 타입으로 마스터 클래스를 가지는 객체를 생성하는 것이 바람직하다.
- <24> 한편, 본 발명의 마스터 클래스는, 객체 내의 데이터를 문자열로 변환하는 문자열 변환 함수를 더 포함하고, 상기 문자열 변환 함수는, 해당 객체 변수를 변수 식별자와, 변수 타입과, 데이터의 개수 및 데이터 내용을 포함하는 문자열로 변환하는 것이 바람직하다.
- <25> 또한, 상기 문자열 변환 함수는, 해당 객체 변수의 변수 타입을 확인하여 마스터 클래스인 경우, 데이터의 개수만큼 재귀적으로 반복 수행되는 것이 효과적이다.
- <26> 여기서, 상기 변환된 문자열은, 각 객체 변수에 대해 변환된 문자열 간에 객체 변수 구분자를 더 포함하여 구성되는 것이 바람직하다.
- <27> 한편, 본 발명의 마스터 클래스는, 객체 내의 데이터를 포함하는 문자열로부터 본래의 객체로 복원하는 객체 복원 함수를 더 포함하고, 상기 객체 복원 함수는, 상기 변환된



문자열에서 분리된 각 객체 변수에 대한 정보에 상기 데이터 저장 함수를 적용하여 복원하는 것이 바람직하다.

<28> 여기서, 상기 객체 복원 함수는, 상기 변환된 문자열에서 분리된 각 객체 변수에 대한 정보 중 데이터의 개수에 따라 메모리를 할당받고, 각 데이터를 순차적으로 저장하는 것이 바람직하다.

<29> 이하에서는 첨부도면을 참조하여 본 발명에 대해 상세히 설명한다.

<30> 도 1은 본 발명의 바람직한 실시예에 따른 마스터 클래스의 구성도이다. 도면에 도시된 바와 같이, 본 발명에 따른 마스터 클래스는, 변수 식별자인 메시지 이름, 변수 타입 정보인 변수 타입, 주소 정보인 주소 정보에 해당되는 속성을 가진다.

<31> 메시지 이름은 문자열 형식의 변수 타입으로 선언되고, 객체 변수의 변수명을 저장할 수 있는 변수 식별자를 지시한다.

<32> 변수 타입은 문자열 형식의 변수 타입으로 선언되고, 변수명으로 지시되는 데이터의 변수 타입 정보를 나타낸다.

<33> 여기서, 변수 타입은 정수, 실수, 문자열, 정수배열, 실수배열, 문자열배열 등과 같이 기존의 변수 타입은 물론이고, 본 발명에서 정의하고 있는 마스터 클래스형, 즉 메시지 객체 클래스의 변수 타입도 수용가능하다.

<34> 기존의 클래스 구성에서, 객체 변수와 해당 객체 변수의 변수 타입은 미리 클래스의 설계시에 결정되어 고정되는 것으로서, 객체 내 함수를 이용하여 각 객체 변수에 저장되어 있는 데이터에 접근하기 위해서, 함수는 객체 변수에 해당되는 변수 타입을 가진 인자를 매개로 한다.

- <35> 그러나, 본 발명의 메세지 객체 클래스는, 함수의 인자로서 객체 변수를 지정하는 변수 식별자와, 그 변수 타입을 넘겨받게 되므로, 객체 변수의 변수 타입이 미리 결정되어 있지 않고 데이터를 해당 객체 변수에 저장하고자 할 때 동적으로 결정되게 된다.
- <36> 주소 정보는 변수 식별자로 지정되는 하나 또는 복수 개의 동일한 변수 타입의 데이터를 지시하는 포인터의 리스트로 구성된다. 따라서, 객체 변수는 단일 변수형 또는 배열(array) 형식의 변수 타입을 모두 수용할 수 있게 되고, 링크드 리스트(linked list) 구조에 의해 크기의 제한을 받지 않는다.
- <37> 링크드 리스트(linked list)는 C에서 쉽게 구현할 수 있는 유용한 데이터 저장방법 중 하나로서, 데이터 간의 결합이 데이터 자체 내에 포함되어 있는 정보에 의해 포인터의 형식으로 정의되는 특성을 가지고 있다. 여기서, 포인터는 다음 데이터의 기억 장소에 대한 정보를 제공하여 데이터 간의 연결 방법을 제공한다. 상기 주소 정보는 포인터의 리스트로서만 구성되기 때문에 실제로 지정될 수 있는 변수 타입에 제한을 받지 않는다.
- <38> 또한, 해당 객체 변수가 적어도 하나의 데이터를 저장할 수 있는 다른 객체 변수를 지시할 수 있도록 마스터 클래스 객체의 포인터를 주소 정보의 변수 타입으로 구성할 수 있도록 함으로써, 상기 객체 변수에 연결된 적어도 하나의 마스터 클래스 객체로 서브리스트를 구성할 수 있도록 한다.
- <39> 위의 과정을 통해서 마스터 클래스로부터 생성된 하나의 객체는 객체 내 객체 변수들이 다른 객체 변수를 지시함으로써 형성되는 객체 간의 연결 관계인 복잡한 트리(tree) 구조를 가지는 데이터 구조(data structure)를 구성할 수 있다.

- <40> 단, 마스터 클래스 객체는 이종의 객체 변수를 수용할 수 있도록 첫 번째 계층에 위치할 수 있는 변수 타입이 마스터 클래스 변수 타입으로 선택된 값을 가지도록 되어 있다. 이러한 구성을 가짐으로써, 마스터 클래스 객체로부터 생성된 하나의 객체의 첫 번째 계층에 속하는 변수 식별자는 첫 번째 계층의 변수의 이름을 나타내는 것이 아니라, 생성된 객체 자체의 이름을 나타내도록 활용할 수 있게 된다.
- <41> 여기서, 계층이란, 객체 내의 객체 변수들이 다른 객체 변수를 지시함으로써 형성되는 객체 간의 연결 관계인 소정의 데이터 구조에 있어서, 연결 관계에 따른 객체가 위치하는 위치 정보를 포함하게 된다.
- <42> 즉, 본 발명에서는, 하나의 객체 변수와 연결된 다른 객체 변수는 서로 다른 계층에 속하게 되는 것이고, 연결 관계의 횟수에 따라 계층을 표현하는 레이어는 증가하는 것으로 정의된다.
- <43> 예를 들어, 생성된 하나의 객체는 레이어0에 있는 것으로 간주하고, 해당 객체에 연결되어 있는 객체 변수들은 레이어1에, 상기 레이어1에 위치하는 객체 변수와 연결되어 있는 객체 변수들은 레이어2에 위치하는 것으로 지정될 수 있다.
- <44> 도 2는 본 발명의 바람직한 실시예에 따른 데이터의 위치 정보를 포함하는 변수 식별자를 나타낸 도면이다. 이는 데이터를 생성하거나 조회하기 위해, 구분자를 활용하여 데이터의 위치 정보를 포함하고 있는 것이다.
- <45> 레이어0의 메시지 이름은 상기에서 언급한 바와 같이, 메시지 객체 자체를 가리키는 변수 식별자에 해당된다.

- <46> 또한, 레이어1의 메세지 이름은 레이어0의 주소 정보가 지시하고 있는 메세지 객체의 변수 식별자에 해당된다. 이와 유사하게, 레이어2의 메세지 이름은 레이어1의 주소 정보가 지시하고 있는 메세지 객체의 변수 식별자이다.
- <47> 나아가, 각 레이어의 메세지 이름, 즉 변수 식별자는 구분자로 분리되어 있다. 본 실시예에서는 구분자로 "."을 이용하고 있으나, 반드시 이에 한정되지는 않으며, "/", "@" 등과 같이 메세지 이름으로 사용되지 않는 문자라면 어느 것이라도 무방하다.
- <48> 여기서, 다음 레이어가 존재하는 해당 레이어의 변수 타입은 상기에서 정의한 메세지 객체의 변수 타입인 것이 바람직하고, 이러한 구성에 의해 각 레이어가 지시하고 있는 데이터들은 서로 계층적으로 연결되게 된다.
- <49> 도 3a는 본 발명의 데이터 저장 함수를 적용하여 형성된 데이터 구조의 예를 나타내는 도면이다.
- <50> 도면에 도시된 바와 같이, 레이어0에는 본 발명에 의해 생성된 객체 자체를 나타내는 변수 식별자인 "메세지 예"와, 다음 레이어인 레이어1에 저장되어 있는 메세지 객체 변수 타입인 "샘플1" 및 "샘플2"의 변수 타입과, 그 각각을 지시하는 주소 정보인 포인터가 저장되어 있는 메세지 객체가 위치하고 있다.
- <51> 레이어1에는 변수 식별자 "샘플1"과, 데이터인 "샘플 문자열1" 및 "샘플 문자열2"의 변수 타입인 문자열과, 그 각각을 지시하는 주소 정보인 포인터가 저장되어 있는 메세지 객체가 위치하고 있다.

- <52> 또한, 레이어1에는 변수 식별자 "샘플2"와, 다음 레이어인 레이어2에 저장되어 있는 메세지 객체 변수 타입인 "샘플3" 및 "샘플4"의 변수 타입과, 그 각각을 지시하는 주소 정보인 포인터가 저장되어 있는 메세지 객체가 위치하고 있다.
- <53> 레이어2에는 변수 식별자 "샘플3"과, 데이터인 "샘플 서브문자열1" 및 "샘플 서브문자열2"의 변수 타입인 문자열과, 그 각각을 지시하는 주소 정보인 포인터가 저장되어 있는 메세지 객체가 위치하고 있다.
- <54> 또한, 레이어2에는 변수 식별자 "샘플4"와, 데이터인 10000 및 20000의 변수 타입인 정수와, 그 각각을 지시하는 주소 정보인 포인터가 저장되어 있는 메세지 객체가 위치하고 있다.
- <55> 도 3b는 도 3a에 나타낸 객체 변수들이 저장하고 있는 데이터 구성 예를 도식적으로 표현한 도면이다. 본 도면에서, 화살표는 포인터를 나타내고 있으며, 실선으로 표시되어 있는 부분은 데이터를 나타내고 있다.
- <56> 한편, 도1에서 보는 바와 같이, 마스터 클래스에는 속성을 저장하기 위한 객체 변수와, 이러한 객체 변수에 접근하여 데이터에 소망하는 작업을 수행하는 복수 개의 함수가 정의되어 있다.
- <57> 본 도면에는, 데이터 저장 함수, 데이터 조회 함수, 문자열 변환 함수 및 객체 복원 함수가 표현되어 있으나, 여기에 한정되지 않고, 선택적으로 이들을 포함할 수 있다.
- <58> 도 4는 본 발명에 따른 데이터 저장 함수의 작동을 보여주는 흐름도이다.
- <59> 본 발명의 데이터 저장 함수는, 변수 식별자, 변수 타입 및 데이터 값을 그 인자로 갖게 된다.

- <60> 먼저, 상기와 같은 메세지 객체 클래스를 정의한다(S1).
- <61> 즉, 객체 변수의 변수명을 선택적으로 저장할 수 있는 변수 식별자와, 상기 객체 변수에 저장될 데이터의 변수 타입을 선택적으로 저장할 수 있는 변수 타입 정보와, 상기 데이터가 저장되어 있는 메모리 주소를 지시하는 주소 정보를 포함할 수 있는 마스터 클래스를 정의한다.
- <62> 생성하고자 하는 위치에 데이터를 생성하기 위해서, 생성하고자 하는 데이터의 데이터 구조상의 위치 정보를 포함하여 변수 식별자를 입력한다(S3). 위치 정보를 포함하고 있는 변수 식별자를 식별자 정보라고 명명하기로 한다.
- <63> 데이터 구조상의 위치 정보는 상기에서 기술한 바와 같이, 데이터가 위치할 수 있는 데이터 구조상의 계층을 나타내는 레이어를 구분하는 구분자를 포함하는 것이 바람직하다.
- <64> 입력된 식별자 정보를 이용하여, 생성하고자 하는 위치에 데이터를 생성한다. 생성하고자 하는 위치에 데이터를 생성하는 단계는 다음과 같다.
- <65> 먼저, 상기 식별자 정보 중 현재 레이어, 예를 들어 레이어k의 메세지 이름에 해당하는 정보를 분리한다(S5).
- <66> 이와 더불어, 상기 식별자 정보에 다음 레이어, 예를 들어 레이어k+1에 해당하는 정보가 있는지 확인한다(S7). 본 발명의 바람직한 실시예에 따르면, 각 레이어는 구분자로 분리되므로, 상기 식별자 정보에 구분자가 있는 경우 다음 레이어에 해당하는 정보가 있는 것으로 판단된다.

- <67> 다음 레이어에 대한 정보가 없는 것으로 판단된 경우, 상기 식별자 정보 중 현재 레이어k의 메시지 이름과 동일한 변수 식별자를 가진 객체 변수가 이미 데이터 구조 내에 존재하는지 여부를 확인한다(S9).
- <68> 상기 식별자 정보 중 현재 레이어k의 메시지 이름과 동일한 변수 식별자를 가진 메시지 객체가 데이터 구조 내에 존재하지 않는 경우, 생성하고자 하는 메시지 이름을 가진 메시지 객체를 생성한다(S11).
- <69> 상기 메시지 객체를 생성하는 것은, 새로운 메시지 객체의 저장 공간을 할당받아 상기 현재 레이어k의 메시지 이름을 변수 식별자에 저장하고, 저장하고자 하는 데이터의 주소 정보를 상기 할당받은 메시지 객체의 주소 정보에 저장하는 것을 포함한다.
- <70> 한편, 상기 식별자 정보 중 현재 레이어k의 메시지 이름과 동일한 메시지 객체가 데이터 구조 내에 존재하는 경우, 저장하고자 하는 데이터의 주소 정보를 상기 메시지 객체의 주소 정보에 추가한다(S13).
- <71> 상기 메시지 객체의 주소 정보에 저장하고자 하는 데이터의 주소 정보를 추가하는 것은, 상기 메시지 객체의 변수 타입에 따라 달라질 수 있다.
- <72> 상기 메시지 객체의 변수 타입이 배열형 변수가 아닌 경우에는, 상기 메시지 객체의 변수 타입을 배열형 변수로 변환하는 것이 바람직하다.
- <73> 상기 식별자 정보에 다음 레이어, 예를 들어 레이어k+1에 해당하는 정보가 있는지 확인한 결과(S7), 다음 레이어에 대한 정보가 있는 것으로 판단된 경우, 상기 식별자 정보 중 현재 레이어k의 메시지 이름과 동일한 메시지 객체가 이미 데이터 구조 내에 존재하는지 여부를 확인한다(S15).

- <74> 상기 식별자 정보 중 현재 레이어k의 메세지 이름과 동일한 메세지 객체가 데이터 구조 내에 존재하는 경우, 다음 레이어k+1로 이동하여(S17), 상기 단계 S5부터 반복 수행한다.
- <75> 한편, 상기 식별자 정보 중 현재 레이어k의 메세지 이름과 동일한 메세지 객체가 데이터 구조 내에 존재하지 않는 경우, 상기 식별자 정보 중 현재 레이어k의 메세지 이름을 변수 식별자로 하는 메세지 객체를 등록한다(S19).
- <76> 여기서, 메세지 객체를 등록하는 것은, 상기 식별자 정보 중 현재 레이어k의 메세지 이름을 변수 식별자로 하는 메세지 객체를 할당받는 것을 포함한다.
- <77> 메세지 객체를 등록한 후, 다음 레이어k+1로 이동하여(S17), 상기 단계 S5부터 반복 수행한다.
- <78> 데이터를 데이터 구조상의 원하는 위치에 생성시키는 상기의 데이터 저장 함수와, 원하는 위치의 데이터 값을 독출하는 데이터 조회 함수는, 데이터에의 접근이라는 측면에서 그 흐름이 유사하다. 각 메세지 객체의 주소 정보를 이용하여 원하는 위치까지 접근한 후, 생성인 경우에는 데이터를 그 위치에 저장하고, 조회인 경우에는 해당 데이터 값을 추출하는 것에서 차이가 있다.
- <79> 상기와 같은 방법에 의해 생성된 데이터 구조에 있어서, 데이터 값을 조회하는 방법은 도 5를 참조하여 설명하기로 한다.
- <80> 본 발명의 데이터 조회 함수는, 데이터 값을 추출하고자 하는 변수 식별자를 그 인자로 가지게 된다.

- <81> 조회하고자 하는 데이터의 데이터 구조상의 위치 정보를 포함하여 변수 식별자를 입력한다(S21).
- <82> 상기 식별자 정보 중 현재 레이어, 예를 들어 레이어k의 메시지 이름에 해당되는 정보를 분리한다(S23).
- <83> 이와 더불어, 상기 식별자 정보에 다음 레이어, 예를 들어 레이어k+1에 해당하는 정보가 있는지 확인한다(S25). 본 발명의 바람직한 실시예에 따르면, 각 레이어는 구분자로 분리되므로, 상기 식별자 정보에 구분자가 있는 경우 다음 레이어에 해당하는 정보가 있는 것으로 판단된다.
- <84> 다음 레이어에 대한 정보가 없는 것으로 판단된 경우, 상기 식별자 정보 중 현재 레이어k의 메시지 이름과 동일한 변수 식별자를 가진 메시지 객체가 이미 데이터 구조 내에 존재하는지 여부를 확인한다(S27).
- <85> 상기 식별자 정보 중 현재 레이어k의 메시지 이름과 동일한 변수 식별자를 가진 메시지 객체가 데이터 구조 내에 존재하지 않는 경우, 조회는 실패로 돌아간다(S29).
- <86> 한편, 상기 식별자 정보 중 현재 레이어k의 메시지 이름과 동일한 변수 식별자를 가진 메시지 객체가 데이터 구조 내에 존재하는 경우, 상기 메시지 객체의 주소 정보를 이용하여 상기 메시지 객체의 데이터를 추출한다(S31).
- <87> 상기 식별자 정보에 다음 레이어, 예를 들어 레이어k+1에 해당하는 정보가 있는지 확인한 결과(S25), 다음 레이어에 대한 정보가 있는 것으로 판단된 경우, 상기 식별자 정보 중 현재 레이어k의 메시지 이름과 동일한 변수 식별자를 가진 메시지 객체가 이미 데이터 구조 내에 존재하는지 여부를 확인한다(S33).

- <88> 상기 식별자 정보 중 현재 레이어k의 메시지 이름과 동일한 변수 식별자를 가진 메시지 객체가 데이터 구조 내에 존재하는 경우, 다음 레이어k+1로 이동하여(S35), 상기 단계 S23부터 반복 수행한다.
- <89> 한편, 상기 식별자 정보 중 현재 레이어k의 메시지 이름과 동일한 변수 식별자를 가진 메시지 객체가 데이터 구조 내에 존재하지 않는 경우, 조회는 실패로 돌아간다(S29).
- <90> 상기와 같은 방법에 의해 생성된 데이터 구조를 가지는 메시지 객체의 데이터들을 하나의 문자열로 변환하는 함수는 도 6을 참조하여 설명하기로 한다.
- <91> 하나의 객체 내의 모든 객체 변수에 저장되어 있는 데이터들, 즉 데이터 구조상의 최상위 메시지 객체와 계층적으로 연결된 복수의 데이터들을 하나의 문자열로 변환하고, 이렇게 변환된 문자열을 다시 계층적 구조로 복원하는 것은 데이터의 전달에 유용할 수 있다.
- <92> 본 발명의 문자열 변환 함수는, 문자열로 변환하고자 하는 메시지 객체의 이름인 식별자를 그 인자로 갖게 된다.
- <93> 먼저, 문자열로 변환하고자 하는 메시지 객체를 지정한다(S41).
- <94> 다음으로, 각 레이어별로 순차적으로 이렇게 지정된 메시지 객체의 식별자에서 현재 레이어에 해당되는 변수 식별자만을 추출하여, 해당 메시지 객체의 메시지 이름, 데이터의 변수 타입, 데이터의 개수 및 데이터의 내용을 추출한다(S43). 상기 메시지 이름, 데이터의 변수 타입 및 데이터의 내용은 해당 정보를 독출하는 것으로 충분하고, 상기 데이

터의 개수는 주소 정보의 개수를 조사하여 추출할 수 있다. 상기 데이터의 내용은 이를 지시하는 주소 정보를 읽어 순차적으로 독출하는 것이 가능하다.

<95> 이렇게 추출된 정보를 이용하여, 해당 메시지 객체의 메시지 이름을 작성하고, 상기의 데이터의 변수 타입, 데이터의 개수 및 데이터의 내용을 하나의 문자열로 변환하여 기존의 문자열에 부가한다(S45).

<96> 해당 메시지 객체의 데이터를 하나의 문자열로 변환하여 기존의 문자열에 부가하기 전에, 메시지 객체의 식별을 가능하도록 하는 구분자를 첨가하는 것이 바람직하다.

<97> 상기 단계 S43 및 S45을 수행함으로써, 데이터 구조 내에 저장되어 있는 데이터는 원래의 데이터 구조로 복원가능하도록 이에 요구되는 정보를 모두 포함할 수 있게 된다.

<98> 해당 메시지 객체의 변수 타입이 메시지 객체인 경우, 데이터의 개수만큼 재귀적으로 반복 수행되도록 하는 것이 바람직한데, 이는 다음과 같은 과정으로 구체화될 수 있다.

<99> 해당 메시지 객체의 변수 타입이 메시지 객체인지 여부를 확인하여(S47), 메시지 객체인 경우에는, 메시지 객체가 지시하고 있는 데이터의 개수만큼 저장된 데이터에 대해 순차적으로 상기 단계 S43 및 S45를 반복수행한다.

<100> 이것은, 상기 메시지 객체가 지시하고 있는 데이터의 개수가 1개인지 여부를 확인하여(S51), 1개인 경우에는 해당 메시지 객체가 지시하고 있는 다음 메시지 객체로 이동하게 되고(S53), 1개 이상인 경우에는 나머지 주소 정보를 스택에 저장하고 최초의 주소 정보가 가리키는 메시지 객체로 이동한다(S53).

- <101> 여기서, 스택은 최후에 저장된 정보가 최선으로 처리되도록 해주는 정보저장 방식에 해당되므로, 현재 메세지 객체의 변수 타입이 메세지 객체이고 복수의 주소 정보를 가지고 있는 경우, 상기 S43 이하의 단계를 재귀적으로 반복 수행하는 효과를 제공한다.
- <102> 즉, 상기 단계 S43 내지 S47은 하나의 객체에 대해서 문자열로 변환하는 과정을 보여주는 것이므로, 문자열로 변환해야 하는 데이터가 여전히 존재하는 경우(S49)에는 상기 단계 S43 이하를 재귀적으로 반복 수행될 수 있다.
- <103> 단계 S41 내지 S55를 수행함으로써, 객체 내의 데이터들에 관한 정보는 하나의 문자열을 형성하게 된다.
- <104> 객체 내의 데이터들을 하나의 문자열로 변환하는 것은, 두 개 이상의 상이한 프로그램이 운용되고 있을 때, 데이터를 다른 프로그램에 전달하는 것이 가능하도록 하기 위해서이다.
- <105> 문자열로 변환된 데이터를 전달받은 프로그램은 상기에서 기술한 객체 내의 데이터 저장 방법을 이용하여 이를 원래의 데이터 구조를 가진 객체로 복원할 수 있다. 원래의 데이터 구조를 가진 객체로 복원하는 함수는, 상기 변환된 문자열에서 분리된 각 객체 변수에 대한 정보에 상기 데이터 저장 함수를 적용하여 복원할 수 있다.
- <106> 즉, 객체 복원 함수는, 상기 변환된 문자열에서 분리된 각 객체 변수에 대한 정보 중 데이터의 개수에 따라 메모리를 할당받고, 각 데이터를 순차적으로 저장하게 되는 것이다.

- <107> 본 발명의 문자열 변환 함수에 의해 변환된 문자열에서 하나의 객체 변수에 대한 정보를 분리하면, 이는 해당 객체 변수의 변수 식별자, 변수 타입, 데이터의 개수 및 데이터 내용을 포함하게 된다.
- <108> 즉, 상기 변수 식별자를 메세지 이름에 저장하고, 변수 타입을 변수 타입 정보에 저장하는 동시에, 데이터의 개수를 읽어 데이터가 저장될 공간을 할당받아, 데이터 내용을 상기 할당받은 저장 공간에 순차적으로 저장하고, 그 주소 정보를 주소 정보에 저장하는 것으로 하나의 객체 변수에 대한 복원은 완료되게 된다.
- <109> 각 객체 변수에 대한 상기와 같은 복원 과정을 반복하여 하나의 객체를 원래의 데이터 구조를 가진 객체로 복원할 수 있다.
- <110> 이러한 구성에 의하여, 다양한 종류의 객체를 생성할 수 있도록 변수 타입이나 변수 개수에 한정되지 않는 새로운 하나의 클래스를 제공함으로써, 클래스 설계 및 프로그램 개발에 드는 시간 및 비용을 절감할 수 있게 된다.

【발명의 효과】

- <111> 이상 설명한 바와 같이, 본 발명에 따르면, 어떠한 종류의 객체라도 생성할 수 있는 마스터 클래스를 제공함으로써, 구현 내용의 변경에 의한 프로그램 인터페이스 환경을 변경하지 않고도 손쉽게 기능을 추가하거나 변경 및 삭제할 수 있게 된다.
- <112> 또한, 서로 다른 두개 이상의 프로그램이 연동하여 운영되는 경우에도 내부적으로 구성된 메세지 객체를 문자열로 변환하고 이를 전달 복원하여 원래와 동일한 메세지 객체를 재구성함으로써, 모든 프로그램에서 표준화된 동일한 인터페이스를 가지는 프로그램으로 구성하는 것이 가능하게 된다.

<113> 따라서, 클래스 설계 및 프로그램 개발에 드는 시간 및 비용을 절감할 수 있게 된다.

【특허청구범위】**【청구항 1】**

임의의 데이터를 저장할 객체 변수 및 상기 객체 변수를 통하여 상기 데이터에 수행되는 복수의 함수를 포함하는 객체로 구성된 객체 지향 프로그램이 기록된 저장 매체에 있어서,

객체 변수의 변수명을 선택적으로 저장할 수 있는 변수 식별자와, 상기 객체 변수에 저장될 데이터의 변수 타입을 선택적으로 저장할 수 있는 변수 타입 정보와, 상기 데이터가 저장되어 있는 메모리 주소를 지시하는 주소 정보를 포함할 수 있는 마스터 클래스와;

상기 마스터 클래스 내에 마련되어, 상기 변수 식별자, 변수 타입 및 데이터 값을 입력받아 상기 객체 변수에 상기 데이터가 저장될 수 있도록 하는 데이터 저장 함수를 포함하는 것을 특징으로 하는 객체 지향 프로그램이 기록된 저장 매체.

【청구항 2】

제1항에 있어서,

상기 변수명은, 문자열로 표현되는 것을 특징으로 하는 객체 지향 프로그램이 기록된 저장 매체.

【청구항 3】

제2항에 있어서,

상기 변수 타입 정보는, 문자열로 표현되고, 해당 객체 변수가 적어도 하나의 데이터를 저장할 수 있는 다른 객체 변수를 지시할 수 있도록 마스터 클래스를 지정할 수 있는 것을 특징으로 하는 객체 지향 프로그램이 기록된 저장 매체.

【청구항 4】

제3항에 있어서,

상기 주소 정보는, 하나 이상의 데이터를 저장할 수 있도록 포인터의 리스트로 구성되는 것을 특징으로 하는 객체 지향 프로그램이 기록된 저장 매체.

【청구항 5】

제4항에 있어서,

상기 데이터 저장 함수에서 입력받는 상기 변수 식별자는, 객체 내의 객체 변수들이 다른 객체 변수를 지시함으로써 형성되는 객체 간의 연결 관계인 소정의 데이터 구조상의 위치 정보를 포함하고 있고, 상기 데이터 저장 함수는 해당 위치의 객체 변수에 상기 데이터를 저장하는 것을 특징으로 하는 객체 지향 프로그램이 기록된 저장 매체.

【청구항 6】

제5항에 있어서,

상기 데이터 구조는 객체 변수간 연결에 따라 레이어가 증가되는 구조이고, 상기 위치 정보는 데이터를 저장하고자 하는 객체 변수에 계층적으로 연결된 상위 레이어에 위치한 다른 객체 변수의 변수 식별자와, 상기 객체 변수들의 변수 식별자를 각각 구분하는 구분자를 포함하는 것을 특징으로 하는 객체 지향 프로그램이 기록된 저장 매체.

【청구항 7】

제6항에 있어서,

상기 데이터 저장 함수는, 상기 데이터 구조상의 해당 위치에 객체 변수가 존재하는 경우, 변수 타입을 정수배열, 문자열배열과 같은 어레이 형식의 변수 타입으로 설정하고, 상기 객체 변수에 상기 데이터를 추가로 저장하는 것을 특징으로 하는 객체 지향 프로그램이 기록된 저장 매체.

【청구항 8】

제7항에 있어서,

상기 데이터 저장 함수는, 상기 위치 정보 중 각 레이어에 위치한 다른 객체 변수가 상기 데이터 구조상에 존재하지 않는 경우, 변수 식별자로 상기 존재하지 않는 객체 변수의 식별자를, 변수 타입으로 마스터 클래스를 가지는 객체를 생성하는 것을 특징으로 하는 객체 지향 프로그램이 기록된 저장 매체.

【청구항 9】

제1항 내지 제8항 중 어느 한 항에 있어서,

상기 마스터 클래스는, 객체 내의 데이터를 문자열로 변환하는 문자열 변환 함수를 더 포함하고, 상기 문자열 변환 함수는, 해당 객체 변수를 변수 식별자와, 변수 타입과, 데이터의 개수 및 데이터 내용을 포함하는 문자열로 변환하는 것을 특징으로 하는 객체 지향 프로그램이 기록된 저장 매체.

【청구항 10】

제9항에 있어서,

상기 문자열 변환 함수는, 해당 객체 변수의 변수 타입을 확인하여 마스터 클래스인 경우, 데이터의 개수만큼 재귀적으로 반복 수행되는 것을 특징으로 하는 객체 지향 프로그램이 기록된 저장 매체.

【청구항 11】

제9항 또는 제10항에 있어서,

상기 변환된 문자열은, 각 객체 변수에 대해 변환된 문자열 간에 객체 변수 구분자를 더 포함하여 구성되는 것을 특징으로 하는 객체 지향 프로그램이 기록된 저장 매체.

【청구항 12】

제9항 내지 11항 중 어느 한 항에 있어서,

객체 내의 데이터를 포함하는 문자열로부터 본래의 객체로 복원하는 객체 복원 함수를 더 포함하고, 상기 객체 복원 함수는, 상기 변환된 문자열에서 분리된 각 객체 변수에 대한 정보에 상기 데이터 저장 함수를 적용하여 복원하는 것을 특징으로 하는 객체 지향 프로그램이 기록된 저장 매체.

【청구항 13】

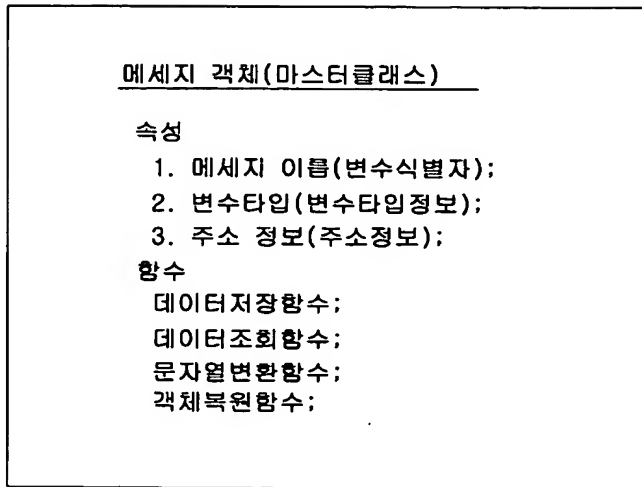
제12항에 있어서,

상기 객체 복원 함수는, 상기 변환된 문자열에서 분리된 각 객체 변수에 대한 정보 중 데이터의 개수에 따라 메모리를 할당받고, 각 데이터를 순차적으로 저장하는 것을 특징으로 하는 객체 지향 프로그램이 기록된 저장 매체.



【도면】

【도 1】



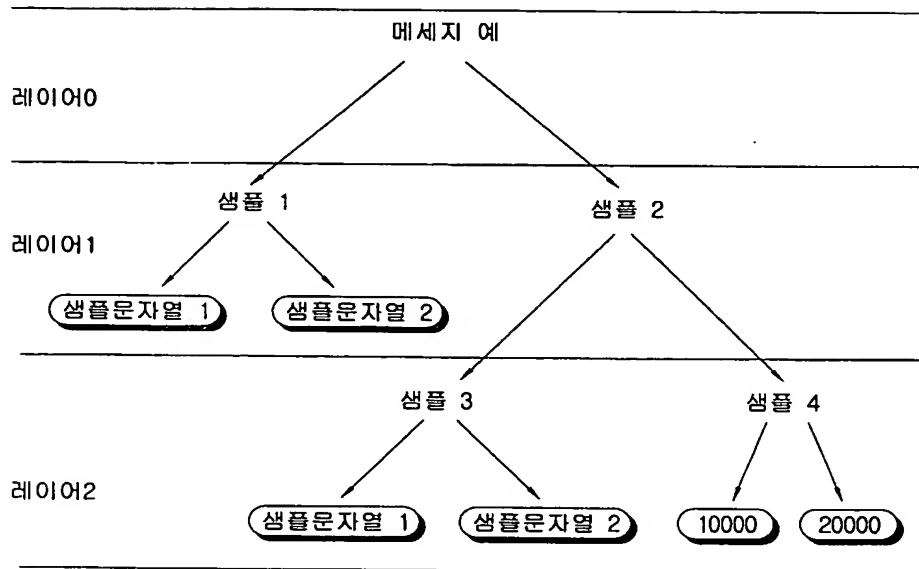
【도 2】

레이어0의 메세지 이름	구분자	레이어1의 메세지 이름	구분자	레이어1의 메세지 이름
이름 0	.	이름 1	.	이름 2

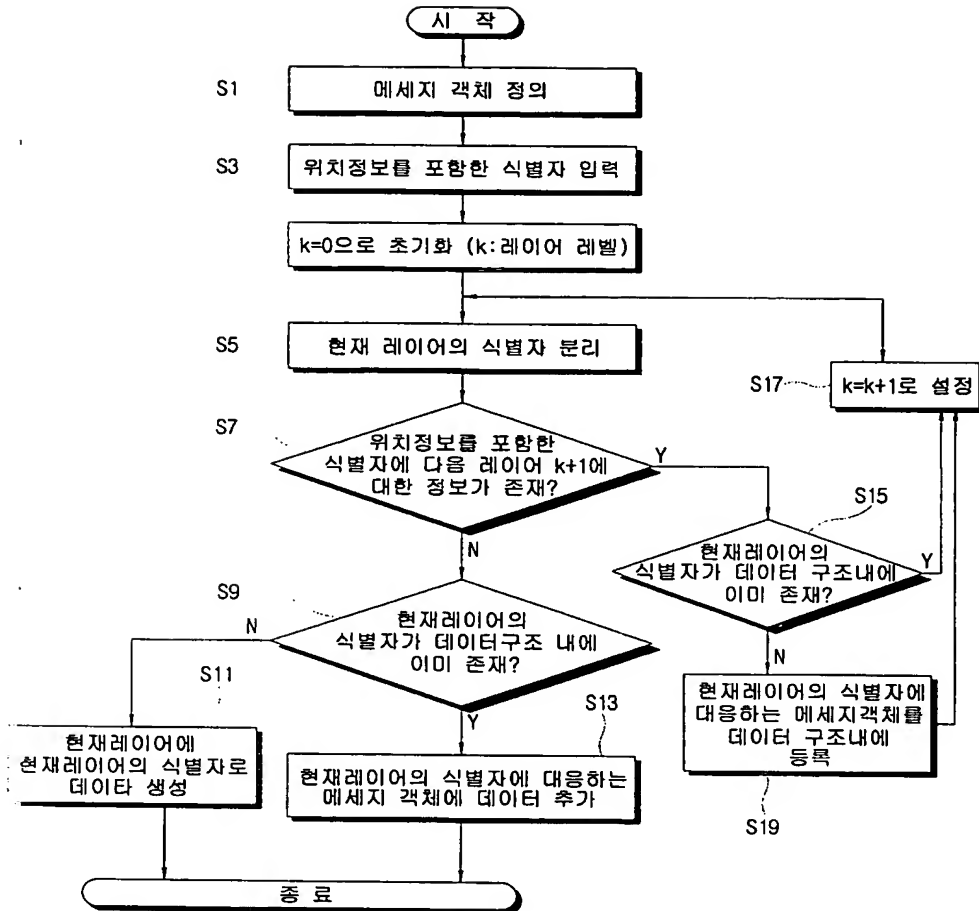
【도 3a】

레이어0		레이어1		레이어2	
메세지이름	"메세지에"				
변수타입	메세지 객체				
주소정보	포인터 메세지 객체	메세지이름	"샘플1"		
		변수타입	문자열		
		주소정보	포인터 "샘플문자열1"		
			포인터 "샘플문자열2"		
	포인터 메세지 객체	메세지이름	"샘플2"		
		변수타입	메세지 객체		
		주소정보	포인터 메세지 객체	메세지이름	"샘플3"
				변수타입	문자열
				주소정보	포인터 "샘플서브문자열1"
					포인터 "샘플서브문자열2"
			포인터 메세지 객체	메세지이름	"샘플4"
				변수타입	정수
				주소정보	포인터 10000
					포인터 20000

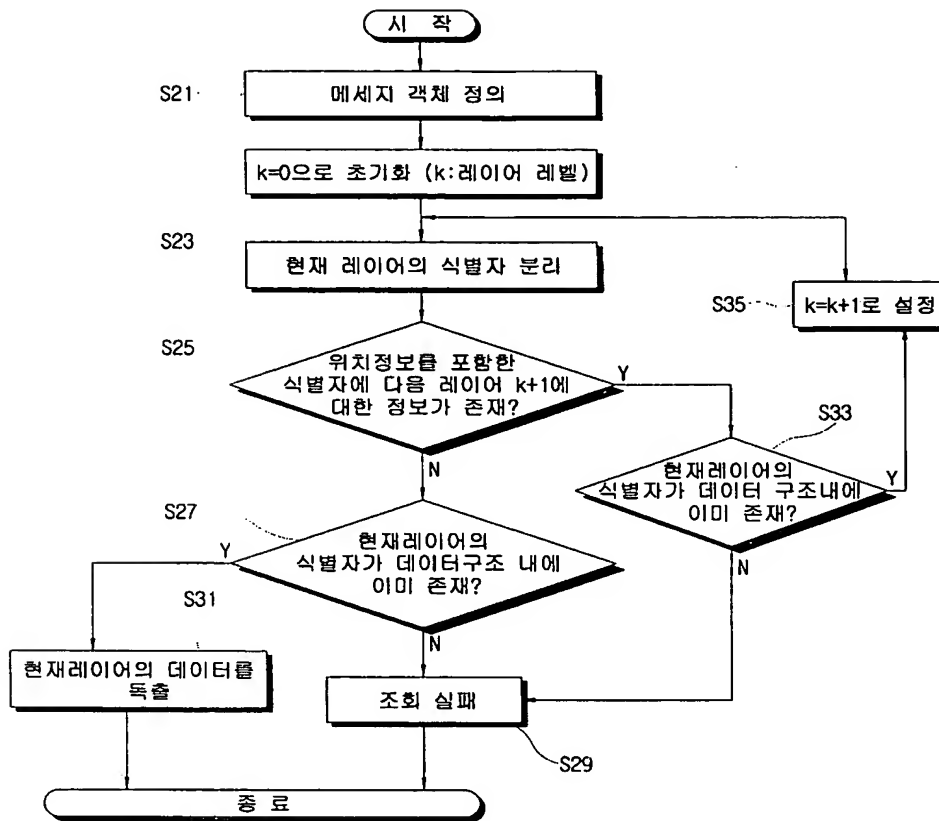
【도 3b】



【도 4】



【도 5】



【도 6】

